Cluster and Cloud Computing Assignment 2 - Australian Social Media Analytics

Background

In development and delivery of non-trivial software systems, working as part of a team is generally (typically!) the norm. This assignment is very much a group project. Students will be put into software teams to work on the implementation of the system described below. These will be teams of up to 5 students. In this assignment, students need to organize their team and their collective involvement throughout. There is no team leader as such, but teams may decide to set up processes for agreeing on the work and who does what. Understanding the dependencies between individual efforts and their successful integration is key to the success of the work and for software engineering projects more generally. If teams have "*issues*", then please let me know asap and I will help resolve them.

Assignment Description

The software engineering activity builds on the lecture materials describing Cloud systems and especially the NeCTAR Research Cloud and its use of OpenStack; on Instagram data (provided); on Twitter APIs, and CouchDB and the kinds of data analytics (e.g. MapReduce) that CouchDB supports as well as data from the Australian Urban Research Infrastructure Network (AURIN – https://portal.aurin.org.au). The focus of this assignment is to harvest tweets from across the cities of Australia on the NeCTAR Research Cloud and undertake a variety of social media data analytics scenarios that tell interesting stories of life in Australian cities **and** importantly how the Twitter data can be used alongside/compared with/augment the data available within the AURIN platform to improve our knowledge of life in the cities of Australia. Teams can download data from the AURIN platform, e.g. as JSON, or use the AURIN openAPI (https://aurin.org.au/aurin-apis/). This data can / should be included into the team's CouchDB database for analysis with Twitter data.

The teams should develop a Cloud-based solution that exploits a multitude of virtual machines (VMs) across the NeCTAR Research Cloud for harvesting tweets through the Twitter APIs (using both the Streaming and the Search API interfaces). The teams should produce a solution that can be run (in principle) across any node of the NeCTAR Research Cloud to harvest and store tweets. Teams have been allocated four medium sized VMs with 8 cores (32Gb memory total) and up to 250Gb of volume storage and 250Gb of object storage. All students have access to the NeCTAR Research Cloud as individual users and can test/develop their applications using their own (small) VM instances. (Remembering that there is no persistence in these small, free and dynamically allocated VMs).

The solution should include a Twitter harvesting application for any/all of the cities of Australia. The teams are expected to have multiple instances of this application running on the NeCTAR Research Cloud together with an associated CouchDB database containing the amalgamated collection of Tweets from the harvester applications. The CouchDB setup may be a single node or adopt a cluster setup. A key aspect of this work is in removing duplicate tweets, i.e. the system should be designed such that duplicate tweets will not arise.

Students may want to explore other social media APIs for collection of data, e.g. Instagram, Foursquare and FlickR, however these are not compulsory to complete the work. A large corpus of Instagram posts will be made available for data analytics, but again teams may decide that they only wish to focus on Twitter data. (See appendix for how to access/download Instagram data). It is noted that social media providers such as Instagram are *evolving* their APIs as well as the policies on access to and use of their data, so bear this in mind is you wish to harvest data from other social media providers. Such data is not compulsory and it is suggested that teams focus on Twitter!

Teams are also expected to develop a range of analytic scenarios, e.g. using the MapReduce capabilities offered by CouchDB for social media analytics and comparing the data with official data from AURIN.

All teams **must** support sentiment analysis, e.g. searching for tweets containing positive sentiments (happy, ecstatic, …), negative sentiments (unhappy, terrible, …) or emoticons like ;o), :o), :), or :o(, >:o(etc and establishing whether people are happier in the morning or in the night time or if there are parts of their cities that are happier than others. Correlating such data with AURIN data should be supported. Teams **should** actively explore more advanced solutions for sentiment analysis rather than simple term searching, e.g. *not happy* is a negative sentiment and *not unhappy* is a neutral sentiment. In addition to this sentiment analysis scenario, teams should explore other scenarios using the AURIN data. Teams are encouraged to be creative here. *A prize will be awarded for the most interesting scenarios identified!* For example teams may look at scenarios such as:

- Which suburb has the most tweeters and does this correlate with what we might expect from the population demographic of the suburb from AURIN, e.g. more young people live in a given area so we might expect a proportionate increase in the number of tweets (assuming young people tweet more)?
- Does the different languages used when tweeting correlate with the cultures we would expect to find in those areas, e.g. more Chinese live in Box Hill in Melbourne hence we would expect to see for tweets tagged as Chinese from those suburbs, or Italians in Carlton etc?
- Are people happier (express more positive sentiment) in areas with more wealth?
- Is there a correlation between crime related tweets and official crime statistics across the suburbs of Melbourne?
- Is there a correlation between alcohol related tweets or crime and locations of places to buy alcohol (bottleshops)?
- Does language use, e.g. vulgar words used in Twitter happen more or less in wealthy or poor areas?

The above are examples – students may decide to create their own analytics based on the data they obtain. Students are not expected to build advanced "general purpose" data analytic services that can support any scenario, but show how tools like CouchDB with targeted data analysis capabilities like MapReduce when provided with suitable inputs can be used to capture the essence of life in Australia. Teams are encouraged to combine twitter data with AURIN data and potentially other data of relevance to the city, e.g. information on weather, sport events, TV shows, visiting celebrities, stock market rise/falls, images from Instagram etc.

A front-end web application is required for visualising these data sets/scenarios.

For the implementation teams are recommended to use a commonly understood language across team members – most likely Java or Python. Information on building and using Twitter harvesters can be found on the web, e.g. see https://dev.twitter.com/ and related links to resources such as Tweepy and Twitter4J. Teams are free to use any pre-existing software systems that they deem appropriate for the analysis. This can include sentiment analysis libraries, gender identification libraries, and machine learning systems as well as front-end Javascript libraries and visualisation capabilities, e.g. Googlemaps.

Error Handling

Issues and challenges in using the NeCTAR Research Cloud for this assignment should be documented. You should describe in detail the limitations of mining twitter content and language processing (e.g. sarcasm). You should outline any solutions developed to tackle such scenarios. Removing duplicates of tweets should be handled. The database may however contain re-tweets. You should demonstrate how you tackled working within the quota imposed by the Twitter APIs through the use of the Cloud.

Final packaging and delivery

You should collectively write a team report on the application developed and include the architecture, the system design and the discussions that lead into the design. You should describe the role of the team members in the delivery of the system and where the team worked well and where issues arose and how they were addressed. The team should illustrate the functionality of the system through a range of scenarios and explain why you chose the specific examples. Teams are encouraged to write this report in the style of a paper than can ultimately be submitted to a conference/journal.

Each team member is also expected to complete a confidential report on their role in the project and the experiences in working with their individual team members. This will be handed in separately to the final team report. (This is not to be used to blame people, but to ensure that all team members are able to provide feedback and to ensure that no team has any member that does nothing!!!).

The length of the team report is not fixed. Given the level of complexity of the assignment and total value of the assignment a suitable estimate is a report in the range of 20-25 pages. A typical report will comprise:

- A description of the system functionalities, the scenarios supported and why, together with graphical results, e.g. pie-charts/graphs of Tweet analysis and snapshots of the web apps/maps displaying certain Tweet scenarios;
- A simple user guide for testing (including system deployment and end user invocation/usage of the systems);
- System design and architecture and how/why this was chosen;
- A discussion on the pros and cons of the NeCTAR Research Cloud and tools and processes for image creation and deployment;
- Teams should also produce a video of their system that is uploaded to YouTube (these videos can last longer than the NeCTAR deployments unfortunately!);
- Reports should also include a link to the source code (github or bitbucket).

It is important to put your collective team details (team, city, names, surnames, student ids) in:

- the head page of the report;
- as a header in each of the files of the software project.

Individual reports describing your role and your teams contributions should be submitted by the PRAZE system.

Implementation Requirements

Teams are expected to use:

- a version-control system such as GitHub or Bitbucket for sharing source code.
- MapReduce based implementations for analytics where appropriate, using CouchDB's built in MapReduce capabilities. You may also consider using Hadoop/Spark for this task if desired.
- The entire system should have scripted deployment capabilities. This means that your team will provide a script, which, when executed, will create and deploy the virtual machines and orchestrate the set up of all necessary software on said machines (e.g. CouchDB, the twitter harvesters, web servers etc.) to create a ready-to-run system. Note that this setup need not populate the database, but demonstrate your ability to orchestrate the necessary software environment on the NeCTAR Research Cloud. Teams should use Ansible (http://www.ansible.com/home) for this task.
- Teams may wish to utilise Docker and technologies but this is not mandatory – especially as the lecture on container technologies takes place mid-way through the assignment.
- The server side of your analytics web application may expose its data to the client through a ReSTful design. Authentication or authorization is NOT required for the web front end.

Teams are also encouraged to describe:

- How fault-tolerant is your software setup? Is there a single point-of-failure?

- Can your application and infrastructure dynamically scale out to meet demand?

Deadline

One copy of the team assignment is to be submitted through the LMS. The zip file must be named with your team, i.e. *<CCC2018-n>.zip*.

Individual reports describing your role and your teams contributions should be submitted via PRAZE on the LMS. These individual reports will be completion of web based forms and not Word/PDF documents etc.

The deadline for submitting the team assignment is: **Thursday 10th May (by 1pm!)**.

Marking

The marking process will be structured by evaluating whether the assignment (application + report) is compliant with the specification given. This implies the following:

- A working demonstration of the Cloud-based solution with dynamic deployment – **25% marks**
- A working demonstration of tweet harvesting and CouchDB utilization for specific analytics scenarios – **25% marks**
- Detailed documentation on the system architecture and design **– 20%**
- Report and write up discussion including pros and cons of the NeCTAR Research Cloud and supporting twitter data analytics – **20% marks**
- Proper handling of the errors and removal of duplicate tweets – **10% marks**

The (confidential) assessment by your peers in your team on PRAZE will be used to weight your individual scores accordingly.

Timeliness in submitting the assignment in the proper format is important. **A 10% deduction per day will be made for late submissions.**

Demonstration Schedule and Venue

The student teams are required to give a presentation (with a few slides) and a demonstration of the working application. This should include the key data analytics scenarios supported as well the design and implementation choices made. Each team has **up to 15 minutes** to present their work. **This will take place on Thursday 10th May (12 teams present) and 17th May (12 teams present). Note that given the numbers of teams this year, not all teams will be able to present – however all teams should be prepared to present on 10th May!!!** I will randomly identify a team on the day (using a random number generator for fairness!!!). Note this is the same day as submission hence the deadline for submission is a hard one!!!

As a team, you are free to develop your system(s) where you are more comfortable with (at home, on your PC/laptop, in the labs...) but obviously the demonstration should work on the NeCTAR Research Cloud.

Appendix – Access to Instagram Data

Note: you do not have to use Instagram data, but if you want to for your scenarios then follow the recipe below. We have been collecting Instagram data on the NeCTAR Research Cloud. There are around 19million posts. We have divided them by location and date of harvesting; the breakdown can be requested with this request (noting that CURL needs to be installed):

```
curl -XGET
"http://45.113.232.90/couchdbro/instagram/_design/instagram/_view/summary?group_level=2" \
--user "readonly:ween7ighai9gahR6"
```

The result of this query is:

```
{"rows":[
{"key":["adelaide",2011],"value":30},
{"key":["adelaide",2012],"value":387},
{"key":["adelaide",2013],"value":907},
{"key":["adelaide",2014],"value":1915},
{"key":["adelaide",2015],"value":5839},
{"key":["adelaide",2016],"value":27220},
{"key":["adelaide",2017],"value":831571},
{"key":["adelaide",2018],"value":417361},
{"key":["brisbane",2010],"value":2},
{"key":["brisbane",2011],"value":40},
{"key":["brisbane",2012],"value":562},
{"key":["brisbane",2013],"value":1552},
{"key":["brisbane",2014],"value":3956},
{"key":["brisbane",2015],"value":10454},
{"key":["brisbane",2016],"value":48278},
{"key":["brisbane",2017],"value":1677850},
{"key":["brisbane",2018],"value":745902},
{"key":["canberra",2011],"value":7},
{"key":["canberra",2012],"value":57},
{"key":["canberra",2013],"value":159},
{"key":["canberra",2014],"value":445},
{"key":["canberra",2015],"value":1136},
{"key":["canberra",2016],"value":5368},
{"key":["canberra",2017],"value":180230},
{"key":["canberra",2018],"value":66237},
{"key":["hobart",2011],"value":1},
{"key":["hobart",2012],"value":4},
{"key":["hobart",2013],"value":24},
{"key":["hobart",2014],"value":63},
{"key":["hobart",2015],"value":134},
{"key":["hobart",2016],"value":867},
{"key":["hobart",2017],"value":28592},
{"key":["hobart",2018],"value":13541},
{"key":["melbourne",2010],"value":12},
{"key":["melbourne",2011],"value":99},
{"key":["melbourne",2012],"value":1216},
{"key":["melbourne",2013],"value":3016},
{"key":["melbourne",2014],"value":7754},
{"key":["melbourne",2015],"value":21634},
{"key":["melbourne",2016],"value":102712},
{"key":["melbourne",2017],"value":3540663},
{"key":["melbourne",2018],"value":1649910},
{"key":["perth",2010],"value":24},
{"key":["perth",2011],"value":66},
{"key":["perth",2012],"value":697},
{"key":["perth",2013],"value":2048},
{"key":["perth",2014],"value":4245},
{"key":["perth",2015],"value":11814},
{"key":["perth",2016],"value":53120},
{"key":["perth",2017],"value":1592720},
{"key":["perth",2018],"value":703355},
{"key":["sydney",2010],"value":12},
{"key":["sydney",2011],"value":159},
{"key":["sydney",2012],"value":1881},
{"key":["sydney",2013],"value":4451},
{"key":["sydney",2014],"value":10567},
{"key":["sydney",2015],"value":28017},
{"key":["sydney",2016],"value":133975},
{"key":["sydney",2017],"value":4588967},
{"key":["sydney",2018],"value":2098862}
]}
```

As an example of using resource to request all Instagram posts for Perth in 2014 and save them in a JSON file, the following request has to be sent:

```
curl "http://45.113.232.90/couchdbro/instagram/_design/instagram/_view/summary" \
-G \
--data-urlencode 'start_key=["perth",2014,1,1]' \
--data-urlencode 'end_key=["perth",2014,12,31]' \
--data-urlencode 'reduce=false' \
--data-urlencode 'include_docs=true' \
--user "readonly:ween7ighai9gahR6" -o /tmp/insta-perth-2014.json
```

The composite key of the summary view is composed of: city name, year, month and day. On average an Instagram post is 1.2K, hence the query above returns a JSON file of about 6MB of disk space.